



The clique-separator graph for chordal graphs[☆]

Louis Ibarra

College of Computing and Digital Media, DePaul University, 243 S. Wabash Ave., Chicago, IL 60604, USA

ARTICLE INFO

Article history:

Received 28 December 2007

Accepted 8 February 2009

Available online 5 April 2009

Keywords:

Chordal graph

Clique tree

Interval graph

Split graph

ABSTRACT

We present a new representation of a chordal graph called the *clique-separator graph*, whose nodes are the maximal cliques and minimal vertex separators of the graph. We present structural properties of the clique-separator graph and additional properties when the chordal graph is an interval graph, proper interval graph, or split graph. We also characterize proper interval graphs and split graphs in terms of the clique-separator graph. We present an algorithm that constructs the clique-separator graph of a chordal graph in $O(n^3)$ time and of an interval graph in $O(n^2)$ time, where n is the number of vertices in the graph.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Chordal graphs and clique trees have an extensive literature and applications in areas such as computational biology, databases, sparse matrix computation, and statistics [2,12,20]. In a clique tree T of a chordal graph G , the nodes of T are the maximal cliques of G and the edges of T correspond to the minimal vertex separators of G . In this paper, we introduce the *clique-separator graph* \mathcal{G} of a chordal graph G . The graph \mathcal{G} is a mixed graph where the nodes are the maximal cliques and minimal vertex separators of G and the (directed) arcs and (undirected) edges represent the containment relations between the maximal cliques and minimal vertex separators of G . The clique-separator graph \mathcal{G} is unique and its structure reflects the structure of G . We show that \mathcal{G} has various structural properties including (a) the edges of \mathcal{G} induce a forest, i.e., deleting the arcs of \mathcal{G} produces a forest, (b) contracting every tree of this forest into a single node yields a directed acyclic graph, (c) for every minimal vertex separator S of G , the vertices of G separated by S are specified by the nodes of \mathcal{G} separated by the node corresponding to S .

When G is an interval graph, the clique-separator graph \mathcal{G} has additional structural properties including (d) every node in \mathcal{G} has at most two predecessors and (e) every tree of the forest (with certain leaves removed) is a path that is the unique clique path on those nodes. When G is a proper interval graph, the clique-separator graph \mathcal{G} has exactly one tree, which is a clique path, and no arcs. When G is a split graph, the clique-separator graph \mathcal{G} has a simple structure centered around one node of \mathcal{G} . (Recall that $\{\text{proper interval graphs}\} \subset \{\text{interval graphs}\} \subset \{\text{chordal graphs}\}$ and that $\{\text{split graphs}\} \subset \{\text{chordal graphs}\}$.)

The clique tree of a chordal graph G does not appear to have structural properties when G belongs to a subclass of chordal graphs. For example, the star $K_{1,n}$ is an interval graph, a split graph, and a ptolemaic graph, and yet any tree on its maximal cliques is a clique tree of $K_{1,n}$. The clique tree is not unique and very different chordal graphs can have isomorphic clique trees. For example, the path on $n + 1$ vertices has exactly one clique tree, which is isomorphic to a clique tree of $K_{1,n}$, which has more than exponentially many clique trees.

The clique-separator graph is different from the *clique graph* in [24], the *weighted clique intersection graph* in [1,2], and the *clique graph* in [10]. The graph in [24] is the intersection graph of the set of maximal cliques of a chordal graph G , i.e., the nodes

[☆] Some of these results were part of the author's Ph.D. thesis at the University of Victoria, BC, Canada.
E-mail address: ibarra@cs.depaul.edu.

are the maximal cliques and two nodes are adjacent exactly when the corresponding cliques have a nonempty intersection. The graph in [1,2] is the same except its edges are weighted by the size of the intersection. The graph in [10] is the same except it contains only the edges corresponding to certain minimal vertex separators; this graph contains every clique tree of G as a subgraph. Each of these is an undirected graph whose nodes are the maximal cliques of G . In contrast, the clique-separator graph's nodes are the maximal cliques *and* the minimal vertex separators of G and it has both arcs and edges.

The clique-separator graph is also different from the *clique laminar tree* for ptolemaic graphs [26], which are the intersection of chordal graphs and distance hereditary graphs. The clique laminar tree is a directed tree where the nodes are the sets of nonempty intersections of the maximal cliques and the arcs represent the containment relations between the sets. In a ptolemaic graph, the sets have the property that for any two sets, either they are disjoint or one is contained in the other, which ensures that the clique laminar tree is a tree. In a chordal graph, neither the maximal cliques nor the minimal vertex separators have this property.

The structural properties of \mathcal{G} when G is an interval graph have been used to develop the first dynamic graph algorithm to recognize interval graphs [15]. A *dynamic graph algorithm* maintains a solution to a graph problem as the graph undergoes a series of small changes, such as single edge insertions or edge deletions. For every change, the algorithm updates the solution faster than recomputing the solution from scratch, i.e., with no previously computed information. The algorithm in [15] maintains the clique-separator graph and supports updates (edge insertions and edge deletions) that produce an interval graph and queries that ask whether a particular update is supported. The algorithm runs in $O(n \log n)$ time per update or query, where n is the number of vertices in the graph. The structural properties of \mathcal{G} when G is a proper interval graph have been used to simplify the algorithm in [15] so that it recognizes proper interval graphs in $O(\log n)$ time per update or query [16]. This algorithm is simpler than the algorithm in [13] and it matches its running time for edge updates and connectivity queries. In comparison, algorithms that recognize an interval graph or proper interval graph from scratch run in $O(m + n)$ time, where m is the number of edges and n is the number of vertices in the graph [3,13]. (All the running times in this paper are worst-case.)

We present the dynamic graph algorithm for interval graphs in a different paper [15] because it requires the introduction of a data structure (the *train tree*) that makes its length substantial. Also, this paper discusses classes of graphs that are unrelated to interval graphs.

In Section 2, we review graph terminology and properties of chordal graphs. In Section 3, we present properties of the clique-separator graph of a chordal graph. In Sections 4–6, we present properties of the clique-separator graph of an interval graph, proper interval graph, and split graph, respectively. In Section 7, we present an algorithm that constructs the clique-separator graph of a chordal graph in $O(n^3)$ time and of an interval graph in $O(n^2)$ time. In Section 8, we consider other subclasses of chordal graphs.

2. Preliminaries

Throughout this paper, let $G = (V, E) = (V(G), E(G))$ be a simple, undirected graph and let $n = |V|$ and $m = |E|$. For a set $S \subseteq V$, the subgraph of G induced by S is $G[S] = (S, E(S))$, where $E(S) = \{\{u, v\} \in E \mid u, v \in S\}$. For a set $S \subset V$, $G - S$ denotes $G[V - S]$. A *clique* of G is a set of pairwise adjacent vertices of G . A *maximal clique* of G is a clique of G that is not properly contained in any clique of G . Fig. 1(a) shows a graph with maximal cliques $\{x, y, z\}$, $\{y, z, w\}$, $\{u, z\}$, and $\{v, z\}$.

Let $S \subset V$. S is a *separator* of G if there exist two vertices that are connected in G and not connected in $G - S$. S is a *minimal separator* of G if S is a separator of G that does not properly contain any separator of G . For $u, v \in V$, S is a *uv-separator* of G if u, v are connected in G and not connected in $G - S$. S is a *minimal vertex separator* of G if for some $u, v \in V$, S is a *uv-separator* of G that does not properly contain any *uv-separator* of G . Every minimal separator is a minimal vertex separator, but not vice versa. Fig. 1(a) shows a graph with minimal separator $\{z\}$ and minimal vertex separators $\{z\}$ and $\{y, z\}$. For $u, v \in V$, S *separates* u, v if S is a *uv-separator* of G . For $X, Y \subset V$, S *separates* X, Y if S is a *uv-separator* of G for every $u \in X$ and $v \in Y$.

A graph is *chordal* (or *triangulated*) if every cycle of length greater than 3 has a *chord*, which is an edge joining two nonconsecutive vertices of the cycle. A graph G is chordal if and only if G has a *clique tree*, which is a tree T on the maximal cliques of G with the *clique intersection property*: for any two maximal cliques K and K' , the set $K \cap K'$ is contained in every maximal clique on the K – K' path in T [5,11,27]. The clique tree is not necessarily unique. Blair and Peyton [2] discuss various properties of clique trees, including the following correspondence between the edges of T and the minimal vertex separators of G . (In [2], G is a connected chordal graph and then the clique intersection property implies that $K \cap K' \neq \emptyset$ for every $\{K, K'\} \in E(T)$.)

Theorem 1 ([14,19]). Let G be a chordal graph with clique tree T . Let $S \neq \emptyset$ be a set of vertices of G .

- (1) S is a minimal vertex separator of G if and only if $S = K \cap K'$ for some $\{K, K'\} \in E(T)$.
- (2) If $S = K \cap K'$ for some $\{K, K'\} \in E(T)$, then S separates $K - S$ and $K' - S$ and moreover, S is a minimal *uv-separator* for any $u \in K - S$ and $v \in K' - S$.

It follows that for any clique tree T of a chordal graph G , the nodes and edges of T correspond to the maximal cliques and minimal vertex separators of G , respectively. A maximal clique corresponds to exactly one node of T and a minimal vertex separator may correspond to more than one edge of T . Furthermore, since a chordal graph G has at most n maximal cliques [9], G has at most $n - 1$ minimal vertex separators.

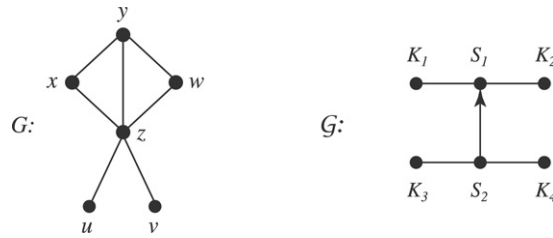


Fig. 1. A chordal graph G and its clique-separator graph \mathcal{G} .

Interval graphs, proper interval graphs, and split graphs are subclasses of chordal graphs and will be defined in the corresponding sections. Golumbic [12] and McKee and McMorris [20] discuss these classes in the context of perfect graphs and intersection graphs, respectively. Brandstädt, Le, and Spinrad [4] also survey these classes and Johnson [17] discusses them with respect to the hardness of problems.

We will use some terms from partially ordered set theory. Let \mathcal{F} be a family of sets. For $S \in \mathcal{F}$, S is a *minimal element* of \mathcal{F} if no $S' \in \mathcal{F}$ satisfies $S' \subset S$, and S is a *maximal element* of \mathcal{F} if no $S' \in \mathcal{F}$ satisfies $S \subset S'$. \mathcal{F} is a *chain* if for any $S, S' \in \mathcal{F}$, we have $S \subseteq S'$ or $S' \subseteq S$. \mathcal{F} is an *antichain* if for any distinct $S, S' \in \mathcal{F}$, we have $S \not\subseteq S'$ and $S' \not\subseteq S$.

3. The clique-separator graph of a chordal graph

3.1. The clique-separator graph

Let G be a chordal graph. The *clique-separator graph* \mathcal{G} of G has the following nodes, (directed) arcs, and (undirected) edges.

- \mathcal{G} has a *clique node* K for each maximal clique K of G .
- \mathcal{G} has a *separator node* S for each minimal vertex separator S of G .
- Each arc is from a separator node to a separator node. \mathcal{G} has arc (S, S') if $S \subset S''$ and there is no separator node S' such that $S \subset S' \subset S''$.
- Each edge is between a separator node and a clique node. \mathcal{G} has edge $\{S, K\}$ if $S \subset K$ and there is no separator node S' such that $S \subset S' \subset K$.

Throughout this paper, we refer to the *vertices* of G and the *nodes* of \mathcal{G} and we use lowercase variables for vertices and uppercase variables for nodes. We identify “maximal clique” and “clique node” and identify “minimal vertex separator” and “separator node”. Fig. 1 shows a chordal graph G and its clique-separator graph \mathcal{G} , which has clique nodes $K_1 = \{x, y, z\}$, $K_2 = \{y, z, w\}$, $K_3 = \{u, z\}$, $K_4 = \{v, z\}$ and separator nodes $S_1 = \{y, z\}$ and $S_2 = \{z\}$.

Let S and S' be distinct separator nodes of \mathcal{G} . If (S, S') is an arc of \mathcal{G} , then S is an *immediate predecessor* of S' and S' is an *immediate successor* of S , denoted $S \rightarrow S'$. If there is a directed path from S to S' in \mathcal{G} , then S is a *predecessor* of S' and S' is a *successor* of S , denoted $S \rightsquigarrow S'$. $S \rightsquigarrow S'$ if and only if $S \subset S'$. Let K be a clique node of \mathcal{G} . If $\{S, K\}$ is an edge of \mathcal{G} , then S is a *neighbor* of K and S is *adjacent* to K and vice versa. The number of neighbors of a node is its *degree*. If S is a neighbor of K or there is a directed path from S to a neighbor of K in \mathcal{G} , then S is a *predecessor* of K and K is a *successor* of S , denoted $S \rightsquigarrow K$. $S \rightsquigarrow K$ if and only if $S \subset K$. The set of immediate predecessors of S , the set of immediate successors of S , and the set of neighbors of K are antichains.

Observation. For every separator node S , there are at least two edges, or two arcs, or an edge and an arc, from S . By Theorem 1(1) $S = K \cap K'$ for some maximal cliques K and K' , and thus $S \rightsquigarrow K$ and $S \rightsquigarrow K'$. If the observation is false, then there is exactly one arc (S, S') from S , where $S' \rightsquigarrow K$ and $S' \rightsquigarrow K'$. But then $S \subset S' \subseteq K \cap K'$, a contradiction.

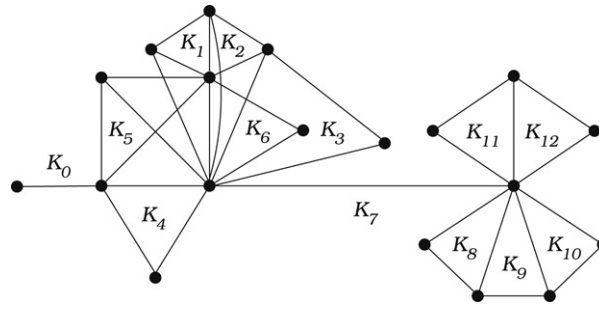
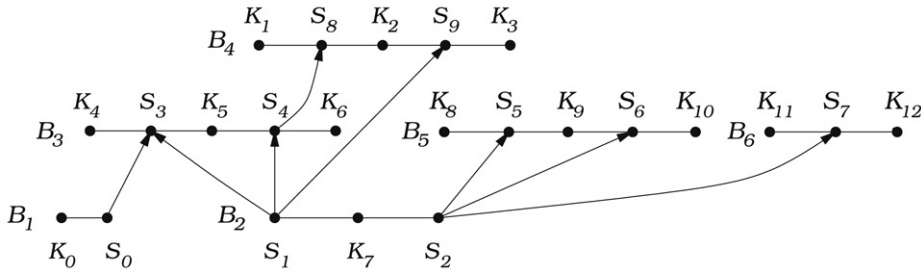
The following is a simple property of the clique-separator graph that follows from the clique intersection property of clique trees.

Lemma 2. Let G be a chordal graph with clique-separator graph \mathcal{G} . For any clique node K and any node $N \neq K$, if $N \cap K \neq \emptyset$, then $N \cap K$ is contained in some neighbor of K in \mathcal{G} .

Proof. Let T be a clique tree of G . We have $N \subseteq K'$ for some maximal clique $K' \neq K$ of G . By the clique intersection property, $N \cap K \subseteq K' \cap K \subseteq K'' \cap K$ for some neighbor K'' of K in T . Since $N \cap K \neq \emptyset$, $K'' \cap K \neq \emptyset$. Then $S = K'' \cap K$ is a separator node of \mathcal{G} by Theorem 1(1). Since $S \subset K$, S is contained in some neighbor of K in \mathcal{G} . \square

For a set of nodes \mathcal{S} of \mathcal{G} , let $V(\mathcal{S}) = \{v \in V \mid v \in N \text{ and } N \in \mathcal{S}\}$. For a subgraph \mathcal{H} of \mathcal{G} , let $V(\mathcal{H}) = \{v \in V \mid v \in N \text{ and } N \text{ is a node of } \mathcal{H}\}$. \mathcal{H} is *connected* if the underlying undirected graph of \mathcal{H} (obtained by replacing every arc with an edge) is connected. \mathcal{G} is connected if and only if G is connected. If a subgraph \mathcal{H} of \mathcal{G} is connected, then $G[V(\mathcal{H})]$ is connected, but the converse is not true in general.

A *box* of \mathcal{G} is a connected component of the undirected graph obtained by deleting the arcs of \mathcal{G} . By Lemma 2, a box of \mathcal{G} contains exactly one clique node K and no separator nodes if and only if K is a complete connected component of G . We

Fig. 2. A chordal graph G .Fig. 3. The clique-separator graph \mathcal{G} of the chordal graph in Fig. 2.

will show that every box is a tree and so we extend the definition of the clique intersection property to a tree T on a set of nodes in the natural way: for any two nodes N and N' of T , the set $N \cap N'$ is contained in every node on the N – N' path in T .

Fig. 2 shows a chordal graph G and Fig. 3 shows its clique-separator graph \mathcal{G} , where separator nodes S_0, S_1, S_2 have size 1, S_8 has size 3, and every other S_i has size 2. The boxes of \mathcal{G} are B_1, B_2, B_3, B_4, B_5 , and B_6 .

The graph \mathcal{G}^c is obtained from \mathcal{G} by contracting each box into a single node and replacing multiple arcs by a single arc. We will denote a box of \mathcal{G} and the corresponding node in \mathcal{G}^c by the same variable. We will show that \mathcal{G}^c is a directed acyclic graph, which means \mathcal{G}^c has a topological sort σ . Given σ , we will assume the boxes of \mathcal{G} are indexed so that $\sigma = (B_1, B_2, \dots, B_{b(\mathcal{G})})$, where $b(\mathcal{G})$ is the number of boxes of \mathcal{G} . Let $\mathcal{G}(\sigma, i)$ be the subgraph of \mathcal{G} induced by the nodes in $B_i, B_{i+1}, \dots, B_{b(\mathcal{G})}$. We will write $\mathcal{G}(\sigma, i)$ as $\mathcal{G}(i)$ if the topological sort σ is clear and we will refer to σ as a topological sort of \mathcal{G} . In Fig. 3, $\sigma = (B_1, B_2, B_3, B_4, B_5, B_6)$ is a topological sort of \mathcal{G} .

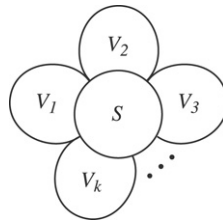
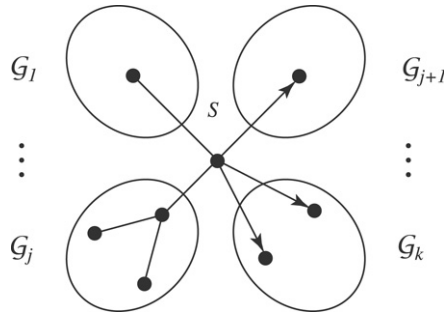
For a separator node S of \mathcal{G} , let $\text{Preds}(S) = \{S' \mid S' = S \text{ or } S' \rightsquigarrow S\}$. We say S divides nodes N and N' if N and N' are contained in the same connected component of \mathcal{G} and in different connected components of $\mathcal{G} - \text{Preds}(S)$. In Fig. 3, S_4 divides K_5 and K_7 and S_5 divides K_8 and K_9 . We use “separates” when referring to G and “divides” when referring to \mathcal{G} .

3.2. The main theorem

Theorem 3. Let G be a chordal graph with clique-separator graph \mathcal{G} and let \mathcal{G} have topological sort σ . Let S be a separator node of \mathcal{G} .

- (1) $G - S$ has connected components G_1, G_2, \dots, G_k and $\mathcal{G} - \text{Preds}(S)$ has connected components $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ such that $k > 1$ and for every $1 \leq i \leq k$, $V(G_i) = V(\mathcal{G}_i) - S$.
- (2) Every box is a tree with the clique intersection property and the set of its separator nodes is an antichain. \mathcal{G}^c is a directed acyclic graph.
- (3) For every $1 \leq i < j \leq b(\mathcal{G})$, if S divides nodes N and N' in $\mathcal{G}(j)$, then S divides N and N' in $\mathcal{G}(i)$.
- (4) For every $1 \leq i < j \leq b(\mathcal{G})$ and every connected component \mathcal{H} of $\mathcal{G}(j)$, at most one separator node of box B_i is a predecessor of any node in \mathcal{H} .
- (5) For every $1 \leq j \leq b(\mathcal{G})$ and every connected component \mathcal{H} of $\mathcal{G}(j)$, $G[V(\mathcal{H})]$ is a connected chordal graph with clique-separator graph \mathcal{H} .

Proof. Since the proof applies to each connected component of G , we assume G is connected. We construct \mathcal{G} inductively. If G is a complete graph, then \mathcal{G} is a single clique node and the theorem is trivially true. Otherwise, let S be a minimal separator of G , which is also a minimal vertex separator of G . Then S is a clique [2,7,12]. Let V_1, V_2, \dots, V_k be the vertex sets of the connected components of $G - S$, $k > 1$. Let G_1, G_2, \dots, G_k be the subgraphs of G induced by $V_1 \cup S, V_2 \cup S, \dots, V_k \cup S$, respectively (Fig. 4).

Fig. 4. Chordal graph G .Fig. 5. Clique-separator graph \mathcal{G} .

Since each of G_1, G_2, \dots, G_k is connected and chordal (because an induced subgraph of a chordal graph is chordal), by induction the theorem holds for each of the corresponding clique-separator graphs $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$. By the choice of S , a separator of any G_i is a separator of G and a minimal vertex separator of any G_i is a minimal vertex separator of G .

Suppose S is a maximal clique of some G_i . Then S is adjacent to some maximal clique K in a clique tree of G_i and $S \cap K \subset S$. Since G_i is connected, $S \cap K \neq \emptyset$. Then by Theorem 1(1), $S \cap K$ is a minimal vertex separator of G_i and thus a minimal vertex separator of G , contradicting the minimality of S . Therefore, S is not a maximal clique of any G_i , or equivalently, S is properly contained in at least one maximal clique of each G_i .

Since S is neither a maximal clique nor a separator of any G_i , S is not a node of any \mathcal{G}_i . Then the \mathcal{G}_i s are node-disjoint. Consider \mathcal{G}_i and \mathcal{G}_j for any $i \neq j$. Since a minimal vertex separator of G_i contains at least one vertex of V_i , \mathcal{G} does not contain an edge $\{S', K'\}$ with S' and K' respectively in \mathcal{G}_i and \mathcal{G}_j . Similarly, \mathcal{G} does not contain an arc (S', S'') with S' and S'' respectively in \mathcal{G}_i and \mathcal{G}_j . Therefore, \mathcal{G} has no edge or arc between a node of \mathcal{G}_i and a node of \mathcal{G}_j . It follows that we may construct \mathcal{G} from the \mathcal{G}_i s by creating separator node S and adding edges and arcs from S to nodes in the \mathcal{G}_i s. We do this in two steps, as follows.

Edge Step: For each i such that S is contained in exactly one maximal clique K of G_i , add an edge between S and K and add no arcs from S . (If S is contained in a minimal vertex separator of G_i , then S is contained in at least two maximal cliques of G_i .) **Arc Step:** For each i such that S is contained in maximal cliques K_1, K_2, \dots, K_l of G_i , $l > 1$, let T_i be a clique tree of G_i . Let M_i be the set of minimal vertex separators of G_i corresponding to edges on a path in T_i between any two of K_1, K_2, \dots, K_l . By the clique intersection property, S is contained in each element of M_i . Add an arc from S to each minimal element of M_i and add no edges from S . This yields the clique-separator graph \mathcal{G} of G .

Without loss of generality, assume the \mathcal{G}_i s are indexed so that the \mathcal{G}_i s in the Edge Step are $\mathcal{G}_1, \dots, \mathcal{G}_j$ and the \mathcal{G}_i s in the Arc Step are $\mathcal{G}_{j+1}, \dots, \mathcal{G}_k$, where $0 \leq j \leq k$ (Fig. 5). Thus, if $1 \leq i \leq j$, we added exactly one edge from S to a clique node of \mathcal{G}_i , and if $j+1 \leq i \leq k$, we added one or more arcs from S to certain separator nodes of \mathcal{G}_i . We now prove each part of the theorem.

1. We defined each G_i to be the subgraph of G induced by $V_i \cup S$, whereas part 1 of the theorem defines each G_i to be the subgraph of G induced by V_i . Since $S = \text{Preds}(S)$, S satisfies part 1. Now let S' be a separator node of some \mathcal{G}_i . By induction, S' satisfies part 1 in \mathcal{G}_i . We must show that S' satisfies part 1 in \mathcal{G} .

Suppose $1 \leq i \leq j$, so that \mathcal{G} has exactly one edge from S to a clique node of \mathcal{G}_i . Then $S \notin \text{Preds}(S')$, which implies $\mathcal{G}_i - \text{Preds}(S')$ and $\mathcal{G} - \text{Preds}(S')$ have the same number of connected components. Now S is a minimal separator (implying every vertex of S is adjacent to at least one vertex of every V_i) and S is a clique. Since $S \not\subset S'$, $\mathcal{G}_i - S'$ and $\mathcal{G} - S'$ have the same number of connected components. It follows that S' satisfies part 1 in \mathcal{G} . Suppose $j+1 \leq i \leq k$, so that \mathcal{G} has arcs from S to separator nodes of \mathcal{G}_i . If $S \notin \text{Preds}(S')$, then $S \not\subset S'$ and the same argument shows that S' satisfies part 1 in \mathcal{G} . If $S \in \text{Preds}(S')$, then $S \subset S'$ and a similar argument shows that S' satisfies part 1 in \mathcal{G} .

2. In the Edge Step, adding an edge between S and one box of each of $\mathcal{G}_1, \dots, \mathcal{G}_j$ causes these boxes to merge into a box B containing S ; since each box is a tree, B is a tree. (If $j = 0$, then B contains only S .) Since S is contained in each of its neighbors in B , it follows that B has the clique intersection property. By the choice of S , B does not contain separator nodes S' and S'' such that $S' \subset S''$. Thus, the set of separator nodes of B is an antichain. Since every arc added in the Arc Step is an arc from S , it follows that \mathcal{G}^c is a directed acyclic graph.

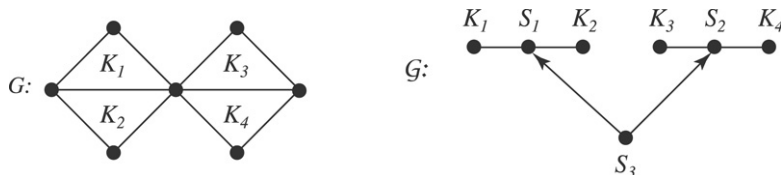


Fig. 6. A chordal graph G and its clique-separator graph \mathcal{G} .

3. Let $\sigma = (B_1, B_2, \dots, B_{b(\mathcal{G})})$. Since \mathcal{G} is connected and not complete, each box B_i has at least one separator node by Lemma 2. A separator node of B_1 has no predecessors and thus it is a minimal separator of G . Therefore, we may assume S is a separator node of B_1 without affecting the proof of parts 1 and 2. Then each of $B_2, \dots, B_{b(\mathcal{G})}$ is a box of some \mathcal{G}_i .

Let σ_i be the restriction of σ to \mathcal{G}_i , $1 \leq i \leq k$. In other words, σ_i is the subsequence of σ that contains the boxes of \mathcal{G}_i ; if $1 \leq i \leq j$, then the box of \mathcal{G}_i contained in B_1 is named B_1 in σ_i . Then σ_i is a topological sort of \mathcal{G}_i . Observe that for $l > 1$, \mathcal{H} is a connected component of $\mathcal{G}(\sigma, l)$ if and only if \mathcal{H} is a connected component of $\mathcal{G}_i(\sigma_i, l)$ for some i . We will use this observation several times in the rest of the proof.

We now show that for every $1 \leq i' < j' \leq b(\mathcal{G})$, if a separator node S' divides nodes N and N' in $\mathcal{G}(\sigma, j')$, then S' divides N and N' in $\mathcal{G}(\sigma, i')$, proving part 3. Suppose S' divides N and N' in $\mathcal{G}(\sigma, j')$. Then N and N' are contained in a connected component of $\mathcal{G}(\sigma, j')$ and by the observation, in a connected component of $\mathcal{G}(\sigma_i, j')$ for some i . It follows that S' divides N and N' in $\mathcal{G}_i(\sigma_i, j')$. By induction, S' divides N and N' in $\mathcal{G}_i(\sigma_i, i')$. If $i' > 1$, then by the observation, every connected component of $\mathcal{G}_i(\sigma_i, i')$ is a connected component of $\mathcal{G}(\sigma, i')$ and thus S' divides N and N' in $\mathcal{G}(\sigma, i')$. Suppose $i' = 1$. Then S' divides N and N' in $\mathcal{G}_i(\sigma_i, 1) = \mathcal{G}_i$ and we must show S' divides N and N' in $\mathcal{G}(\sigma, 1) = \mathcal{G}$.

If $1 \leq i \leq j$, then the definition of the Edge Step implies that S' divides N and N' in \mathcal{G} . If $j + 1 \leq i \leq k$, then suppose \mathcal{G} has arcs from S to separator nodes S_1 and S_2 in different connected components of $\mathcal{G}_i - \text{Preds}(S')$. (If \mathcal{G} has no such arcs, then S' divides N and N' in \mathcal{G} .) Then $S \subset S_1$ and $S \subset S_2$. Now if there is a vertex $v \in S - S'$, then $v \in S_1 - S'$ and $v \in S_2 - S'$. But S' divides S_1 and S_2 in \mathcal{G}_i and by part 1, S' separates $S_1 - S'$ and $S_2 - S'$ in \mathcal{G}_i , a contradiction. Therefore, $S \subset S'$. Then the definition of the Arc Step implies that $S \rightsquigarrow S'$ in \mathcal{G} and thus $S \in \text{Preds}(S')$. Then every connected component of $\mathcal{G}_i - \text{Preds}(S')$ is a connected component of $\mathcal{G} - \text{Preds}(S')$, which implies S' divides N and N' in \mathcal{G} .

4. Let \mathcal{H} be a connected component of $\mathcal{G}(\sigma, j')$, $1 < j' \leq b(\mathcal{G})$. By the observation, \mathcal{H} is a connected component of $\mathcal{G}_i(\sigma_i, j')$ for some i . Consider the boxes of \mathcal{G} . For every box $B_{i'}$ of \mathcal{G} , $1 < i' < j'$, if $B_{i'}$ is a box of \mathcal{G}_i , then by induction, $B_{i'}$ contains at most one predecessor of a node in \mathcal{H} , and otherwise, $B_{i'}$ contains no predecessor of a node in \mathcal{H} . Lastly, B_1 contains at most one predecessor of a node in \mathcal{H} by induction (if $1 \leq i \leq j$) or by the definition of the Arc Step (if $j + 1 \leq i \leq k$).

5. Let \mathcal{H} be a connected component of $\mathcal{G}(\sigma, j')$, $1 \leq j' \leq b(\mathcal{G})$. If $j' = 1$, then $\mathcal{H} = \mathcal{G}$ and $G[V(\mathcal{H})] = G$ and we are done. Suppose $j' > 1$. By the observation, \mathcal{H} is a connected component of $\mathcal{G}_i(\sigma_i, j')$ for some i . By induction, $G_i[V(\mathcal{H})]$ is a connected chordal graph with clique-separator graph \mathcal{H} . Since $V(\mathcal{H}) \subseteq V(\mathcal{G}_i) = V(G_i)$, it follows that $G[V(\mathcal{H})] = G_i[V(\mathcal{H})]$. Hence, $G[V(\mathcal{H})]$ is a connected chordal graph with clique-separator graph \mathcal{H} . \square

The Main Theorem 5 is not used in the rest of this paper, but it is necessary for [15]. Notice that $\mathcal{G}(j)$ is not always the clique-separator graph of the chordal graph $G[V(\mathcal{G}(j))]$. In Fig. 6, for example, $\mathcal{G}(2)$ is not the clique-separator graph of $G[V(\mathcal{G}(2))]$.

We extend the definition of divides to sets of nodes S and S' of \mathcal{G} as follows: S divides S and S' if $S - \text{Preds}(S)$ and $S' - \text{Preds}(S)$ are contained in the same connected component of \mathcal{G} and in different connected components of $\mathcal{G} - \text{Preds}(S)$. We extend the definition of divides to subgraphs \mathcal{H} and \mathcal{H}' of \mathcal{G} in the same way. We need one more definition: S strongly divides nodes N and N' if $S \rightsquigarrow N$ and $S \rightsquigarrow N'$ and S divides N and N' . In Fig. 3, S_4 divides K_5 and K_7 but does not strongly divide K_5 and K_7 , whereas S_1 strongly divides K_5 and K_7 .

Corollary 4. Let G be a chordal graph with clique-separator graph \mathcal{G} . Let S be a separator node of \mathcal{G} .

- (1) Let N and N' be nodes of \mathcal{G} . Then S divides N and N' if and only if S separates $N - S$ and $N' - S$.
Let S and S' be sets of nodes of \mathcal{G} . Then S divides S and S' if and only if S separates $V(S) - S$ and $V(S') - S$.
Let \mathcal{H} and \mathcal{H}' be subgraphs of \mathcal{G} . Then S divides \mathcal{H} and \mathcal{H}' if and only if S separates $V(\mathcal{H}) - S$ and $V(\mathcal{H}') - S$.
- (2) If S has distinct neighbors N and N' , or S has neighbor N and successor N' , then S strongly divides N and N' .
- (3) S strongly divides nodes N and N' if and only if $S = N \cap N'$ and S separates $N - S$ and $N' - S$.
- (4) S strongly divides clique nodes K and K' if and only if $S = K \cap K'$ for some edge $\{K, K'\}$ in a clique tree of G .

Proof. (1) By the Main Theorem 1.

- (2) Let B_i be the box containing S in some topological sort of \mathcal{G} . By the Main Theorem 4, S divides N and N' in $\mathcal{G}(i)$. If $i = 1$, then $\mathcal{G}(i) = \mathcal{G}$, and if $i > 1$, then by the Main Theorem 3, S divides N and N' in \mathcal{G} . Since $S \rightsquigarrow N$ and $S \rightsquigarrow N'$, it follows that S strongly divides N and N' in \mathcal{G} .
- (3) (\Rightarrow) By part 1, S separates $N - S$ and $N' - S$. Since $S \rightsquigarrow N$ and $S \rightsquigarrow N'$, it follows that $S \subseteq N \cap N'$. If $S \subset N \cap N'$, then some vertex is in both $N - S$ and $N' - S$, a contradiction. Thus, $S = N \cap N'$. (\Leftarrow) By part 1, S divides N and N' . Since $S \subseteq N \cap N'$, it follows that $S \rightsquigarrow N$ and $S \rightsquigarrow N'$.

- (4) (\Leftarrow) By Theorem 1(2), S separates $K - S$ and $K' - S$. By part 3, S strongly divides K and K' . (\Rightarrow) Let T be any clique tree of G . If $\{K, K'\}$ is an edge of T , we are done. Otherwise, let $K = K_1, K_2, \dots, K_j = K'$ be the $K-K'$ path in T , where $j \geq 3$. Since T has the clique intersection property, $S \subset K_i$ and thus $S \rightsquigarrow K_i$ for every $1 \leq i \leq j$. Then K_1, K_2, \dots, K_j are in the same connected component of \mathcal{G} . Now S divides K_1 and K_j . Then K_1 and K_j are in different connected components of $\mathcal{G} - \text{Preds}(S)$ and for some $1 \leq i \leq j-1$, K_i and K_{i+1} are in different connected components of $\mathcal{G} - \text{Preds}(S)$. Then S divides K_i and K_{i+1} and thus S strongly divides K_i and K_{i+1} . By part 3, $S = K_i \cap K_{i+1} = K_1 \cap K_j$. Then the tree obtained from T by deleting edge $\{K_i, K_{i+1}\}$ and adding edge $\{K_1, K_j\}$ is a clique tree of G . (A tree is a clique tree of G if and only if it is a maximum spanning tree of the *weighted clique intersection graph* in [1,2]. In this graph, the nodes are the maximal cliques of G and the graph has edge $\{K, K'\}$ with weight $|K \cap K'|$ if and only if $K \cap K' \neq \emptyset$.) \square

Remark. Corollary 4(4) implies that for any connected chordal graph G with clique nodes K and K' , G has a clique tree with edge $\{K, K'\}$ if and only if \mathcal{G} has a separator node that strongly divides K and K' . (We need G to be connected because if K and K' are in different connected components of G , then G has a clique tree with edge $\{K, K'\}$ but no separator node equals $K \cap K' = \emptyset$.)

4. The clique-separator graph of an interval graph

An *interval graph* is a graph where each vertex can be assigned an interval on the real line so that two vertices are adjacent if and only if their assigned intervals intersect; such an assignment is an *interval representation*. A graph is an interval graph if and only if it has a *clique path*, which is a clique tree that is a path [9]. Interval graphs are a proper subclass of chordal graphs. Since interval graphs model many problems involving linear arrangements, interval graphs have as many applications as chordal graphs. Application areas include archeology, computational biology, file organization, partially ordered sets, and psychology [4,12,20].

In this section, we present properties of the clique-separator graph \mathcal{G} when G is an interval graph. A path P on a set of nodes of \mathcal{G} (not necessarily a subgraph of \mathcal{G}) is a *c.i.p.* path if P has the clique intersection property. We will use the following characterization, which is straightforward to prove from the clique path characterization. Let G be a chordal graph with clique-separator graph \mathcal{G} . Then G is an interval graph if and only if there is a *c.i.p.* path on the nodes of \mathcal{G} .

For a clique tree T of G , let $S_T(K) = \{K \cap K' \mid \{K, K'\} \in E(T) \text{ and } K \cap K' \neq \emptyset\}$, which is a set of minimal vertex separators by Theorem 1(1). In the following lemma, part 4 is a property of clique trees that follows from part 1.

Lemma 5. Let G be an interval graph with clique-separator graph \mathcal{G} .

- (1) For every node, the set of its predecessors can be partitioned into at most two chains.
- (2) Every clique node has at most two neighbors. Every separator node has at most two immediate predecessors.
- (3) There are $O(n)$ nodes, edges and arcs in \mathcal{G} .
- (4) For every maximal clique K and clique tree T of G , $S_T(K)$ can be partitioned into at most two chains.

Proof. (1) By the characterization, there is a *c.i.p.* path that contains every node of \mathcal{G} . Let N be any node and let S be the set of its predecessors. If S contains an antichain $\{S_1, S_2, S_3\}$, then since $S_1 \subset N, S_2 \subset N, S_3 \subset N$, there is no *c.i.p.* path containing S_1, S_2, S_3 , and N , a contradiction. Thus, (S, \subseteq) is a partially ordered set whose largest antichain has size at most 2. By Dilworth's Theorem, which states that the size of the largest antichain is equal to the size of the smallest chain decomposition, S can be partitioned into at most two chains.

- (2) By part 1.
- (3) Since G is a chordal graph with at most n maximal cliques and $n-1$ minimal vertex separators, \mathcal{G} has at most $2n-1$ nodes. By the Main Theorem 2, \mathcal{G} has at most $2n-2$ edges. By part 2, \mathcal{G} has at most $2n-2$ arcs.
- (4) By part 1 because the elements of $S_T(K)$ are predecessors of K in \mathcal{G} . \square

A path P on a set of nodes of \mathcal{G} (not necessarily a subgraph of \mathcal{G}) is *constraining* if P is the unique *c.i.p.* path on the set of nodes in P . For paths P and Q on sets of nodes of \mathcal{G} , P is a *subsequence* of Q if, when the paths are viewed as sequences, P or its reverse is a subsequence of Q . Then a constraining path P is a subsequence of every *c.i.p.* path that contains the nodes of P .

Lemma 6. Let G be an interval graph with clique-separator graph \mathcal{G} . Let B be a box of \mathcal{G} .

- (1) Every path in B between two separator nodes is a constraining path.
- (2) Every separator node of B has at most two neighbors that are internal nodes of B .

Proof. By the Main Theorem 2, every path contained in B is a *c.i.p.* path and the set of separator nodes in B is an antichain. Observe that if B contains a separator node S and a clique node K such that $S \subset K$, then S is a neighbor of K , or else $S \subset S'$ for some neighbor S' of K , a contradiction.

- (1) Let $(S_1, K_1, S_2, K_2, S_3)$ be a path contained in B . Then $\{S_1, S_2, S_3\}$ is an antichain. Since $S_1 \subset K_1$ and $S_2 \subset K_1$, it follows that (S_1, K_1, S_2) is a constraining path. By the observation, $S_1 \not\subset K_2$. Then (S_1, K_1, S_2, K_2) and (S_1, K_1, K_2, S_2) are the only *c.i.p.* paths on these four nodes. Since $S_3 \subset K_2$, it follows that $(S_1, K_1, S_2, K_2, S_3)$ is a constraining path. Applying induction with a similar argument shows that every path in B between two separator nodes is a constraining path.

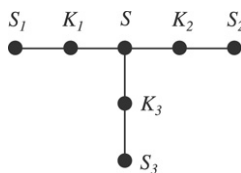


Fig. 7. A separator node S with three neighbors that are internal nodes.

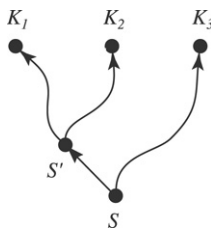


Fig. 8. Proof of Theorem 8.

- (2) Let S be a separator node with three neighbors that are internal nodes of box B (Fig. 7). Then B contains paths (S, K_1, S_1) , (S, K_2, S_2) , (S, K_3, S_3) and $\{S, S_1, S_2, S_3\}$ is an antichain. By part 1, (S_1, K_1, S, K_2, S_2) is a constraining path and by the observation, $S_1 \not\subset K_3$ and $S_2 \not\subset K_3$. It follows that $P = (S_1, K_1, K_3, K_2, S_2)$ is a constraining path. Since $S_3 \subset K_3$, there is no c.i.p. path containing S_3 and the nodes in P . But by the characterization, there is a c.i.p. path containing S_3 and the nodes in P , a contradiction. \square

A *leaf* clique node is a clique node that has degree 1, or equivalently, a clique node that is a leaf of some box. The *body* of a box B , denoted $body(B)$, is the subgraph of B obtained by deleting the leaf clique nodes of B . Lemmas 5 and 6 yield the following theorem, which summarizes the properties of \mathcal{G} when G is an interval graph.

Theorem 7. Let G be an interval graph with clique-separator graph \mathcal{G} . Then for every box B , $body(B)$ is a constraining path whose endpoints are separator nodes. Moreover, every separator node has at most two immediate predecessors and there are $O(n)$ nodes, edges and arcs in \mathcal{G} .

Theorem 7 is used by the dynamic graph algorithm for interval graphs in [15] to decide whether G is an interval graph. The algorithm requires additional results since the condition in Theorem 7 is necessary but not sufficient for G to be an interval graph.

5. The clique-separator graph of a proper interval graph

A *proper interval graph* (or *unit interval graph*) is an interval graph with an interval representation where no interval is properly contained in another. Proper interval graphs are a proper subclass of interval graphs. For example, $K_{1,3}$ (the graph consisting of one vertex adjacent to three nonadjacent vertices) is an interval graph but not a proper interval graph. In fact, an interval graph is a proper interval graph if and only if it has no induced subgraph isomorphic to $K_{1,3}$ [23]. An *independent set* of G is a set of pairwise nonadjacent vertices.

The next theorem presents the properties of \mathcal{G} when G is a proper interval graph. If G is not connected, the theorem can be applied to each of its connected components.

Theorem 8. Let G be a connected chordal graph with clique-separator graph \mathcal{G} . Then G is a proper interval graph if and only if \mathcal{G} has exactly one box and this box is a path P such that there do not exist vertices $u \in K_1 \cap K_3$ and $v \in K_2 - (K_1 \cup K_3)$ where (K_1, K_2, K_3) is a subsequence of P .

Proof. (\Rightarrow) Suppose \mathcal{G} has an arc (S, S') . By Theorem 1(1) and Corollary 4(4), there are clique nodes K_1 and K_2 such that S' strongly divides K_1 and K_2 in \mathcal{G} . Then $S' \rightsquigarrow K_1$ and $S' \rightsquigarrow K_2$ and thus K_1 and K_2 are in one connected component of $\mathcal{G} - \text{Preds}(S)$ (Fig. 8). Then by Corollary 4(4), $S \rightsquigarrow K_3$ for some clique node K_3 such that $\{K_1, K_2\}$ and K_3 are in different connected components of $\mathcal{G} - \text{Preds}(S)$. Then S divides $\{K_1, K_2\}$ and K_3 . Let $y_1 \in K_1 - S'$, $y_2 \in K_2 - S'$, $y_3 \in K_3 - S$. By Corollary 4(1), S' separates y_1 and y_2 and S separates $\{y_1, y_2\}$ and y_3 . Then $\{y_1, y_2, y_3\}$ is an independent set of G . Now any $x \in S \subset S'$ is adjacent to each of y_1, y_2, y_3 , which means G has an induced $K_{1,3}$, a contradiction. Thus, \mathcal{G} has no arc. Since G is connected, \mathcal{G} has exactly one box.

Suppose S is a separator node with three neighbors K_1, K_2, K_3 . Let $y_i \in K_i - S$, $i = 1, 2, 3$. By Corollary 4(1) and 4(2), S pairwise separates y_1, y_2, y_3 . Then $\{y_1, y_2, y_3\}$ is an independent set of G . Since any $x \in S$ is adjacent to each of y_1, y_2, y_3 , G has an induced $K_{1,3}$, a contradiction. Thus, every separator node has at most two neighbors. Since every clique node has at most two neighbors (Lemma 5(2)), the box of \mathcal{G} is a path P .

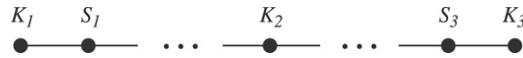


Fig. 9. Proof of Theorem 8.

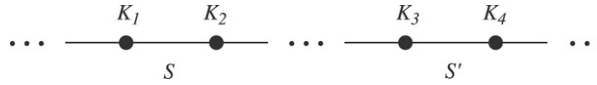


Fig. 10. Proof of Lemma 9.

Suppose there exist vertices $u \in K_1 \cap K_3$ and $v \in K_2 - (K_1 \cup K_3)$ where (K_1, K_2, K_3) is a subsequence of P . Let S_1 and S_3 be the neighbors of K_1 and K_3 that are on the K_1 – K_3 path in P , respectively (Fig. 9). Let $x \in K_1 - S_1$ and $y \in K_3 - S_3$. Since $v \notin K_1 \cup K_3$, it follows that $v \notin S_1 \cup S_3$. By Corollary 4(1), S_1 separates x and v , S_3 separates v and y , and S_1 (or S_3) separates x and y . Then $\{x, v, y\}$ is an independent set of G . Since $u \in K_1 \cap K_3$ and P has the clique intersection property, u is adjacent to each of x, v, y . Then G has an induced $K_{1,3}$, a contradiction.

(\Leftarrow) Since \mathcal{G} has exactly one box and this box is a path P , deleting the separator nodes from P yields a clique path of G . Then G is an interval graph. Suppose G is not a proper interval graph. Then G contains vertices u, v_1, v_2, v_3 such that u adjacent to each of v_1, v_2, v_3 and $\{v_1, v_2, v_3\}$ is an independent set of G . Then $v_1 \in K_1, v_2 \in K_2, v_3 \in K_3$ for distinct clique nodes K_1, K_2, K_3 of P . Without loss of generality, assume that (K_1, K_2, K_3) is a subsequence of P and that K_1 and K_3 are as close as possible on P . Let S_1 and S_3 be the neighbors of K_1 and K_3 on the K_1 – K_3 path in P , respectively (Fig. 9). By Corollary 4(1), S_1 separates $K_1 - S_1$ and $K_2 - S_1$. Since $v_1 \in K_1 - S_1$ and $v_2 \in K_2 - S_1$ and u is adjacent to v_1 and v_2 , it follows that $u \in S_1$. The symmetric argument shows that $u \in S_3$, which means $u \in K_1 \cap K_3$. Now v_2 is adjacent to neither v_1 nor v_3 and thus $v_2 \notin K_1 \cup K_3$. Then $v_2 \in K_2 - (K_1 \cup K_3)$, a contradiction. Hence, G is a proper interval graph. \square

Theorem 8 characterizes proper interval graphs in terms of the clique-separator graph; its corollary characterizes proper interval graphs in terms of the clique tree. The corollary requires the following lemma, which references Theorem 1(1).

Lemma 9. Let G be a connected chordal graph with clique tree T . Then T is the unique clique tree of G if and only if the family of minimal vertex separators corresponding to the edges of T is an antichain.

Proof. We prove the contrapositive of both directions. Since G is connected, the clique intersection property implies that $K \cap K' \neq \emptyset$ for every edge $\{K, K'\} \in E(T)$.

(\Leftarrow) Suppose G has distinct clique trees T and T' . Then there are maximal cliques K and K' that are adjacent in T' and not adjacent in T . Then $S = K \cap K'$ is a minimal vertex separator. Now the K – K' path in T contains (at least) two edges, which correspond to minimal vertex separators S' and S'' . Since T has the clique intersection property, $S \subseteq S'$ and $S \subseteq S''$. We claim that the family of minimal vertex separators corresponding to the edges of T is not an antichain. If $S' = S''$, then the claim holds, and if $S' \neq S''$, then $S \subset S'$ or $S \subset S''$, and again the claim holds.

(\Rightarrow) Suppose the family of minimal vertex separators corresponding to the edges of T is not an antichain. Then T contains distinct edges $\{K_1, K_2\}$ and $\{K_3, K_4\}$ such that $S = K_1 \cap K_2 \subseteq K_3 \cap K_4 = S'$. Without loss of generality, assume that (K_1, K_2, K_3, K_4) is a subsequence of a path contained in T ; we may have $K_2 = K_3$ (Fig. 10). Let T' be the tree obtained from P by deleting edge $\{K_1, K_2\}$ and adding edge $\{K_1, K_4\}$. We claim that T' has the clique intersection property. To show this, let K and K' be maximal cliques such that the K – K' path in T' contains edge $\{K_1, K_4\}$. Then the K – K' path in T contains edge $\{K_1, K_2\}$, which implies that $K \cap K' \subseteq S \subseteq S'$. It follows that the claim holds and thus T and T' are distinct clique trees of G . \square

Corollary 10. Let G be a connected chordal graph. Then G is a proper interval graph if and only if G has a unique clique tree and this tree is a path P such that there do not exist vertices $u \in K_1 \cap K_3$ and $v \in K_2 - (K_1 \cup K_3)$ where (K_1, K_2, K_3) is a subsequence of P .

Proof. (\Rightarrow) Let P be the box of \mathcal{G} satisfying the conditions of Theorem 8. Then deleting the separator nodes of P yields a clique path P' of G , which is a clique tree of G . By the Main Theorem 2, the family of minimal vertex separators corresponding to the edges of P' is an antichain. By Lemma 9, P is the unique clique tree of G . The rest of the statement holds because it is the same in Theorem 8 and in this corollary.

(\Leftarrow) By Lemma 9, the family of minimal vertex separators corresponding to the edges of P is an antichain. Let P' be the path obtained from P by inserting the minimal vertex separator $K \cap K'$ between every consecutive pair K and K' of maximal cliques in P . Since the inserted sets form an antichain, each $K \cap K'$ is not properly contained in any minimal vertex separator or any maximal cliques other than K and K' . Therefore, P' is a box of \mathcal{G} . Moreover, since P' contains every node of \mathcal{G} , P' is the only box of \mathcal{G} . By Theorem 8, G is a proper interval graph. \square

6. The clique-separator graph of a split graph

A *split graph* is a graph whose vertex set can be partitioned into a clique K and an independent set I , where either set may be empty and there may be edges between vertices in K and vertices in I . A graph G is a split graph if and only if G and its complement \bar{G} are chordal [8]. (The complement of $G = (V, E)$ is $\bar{G} = (V, \bar{E})$ where $\bar{E} = \{\{u, v\} \mid u, v \in V \text{ and } u \neq v, uv \notin E\}$.)

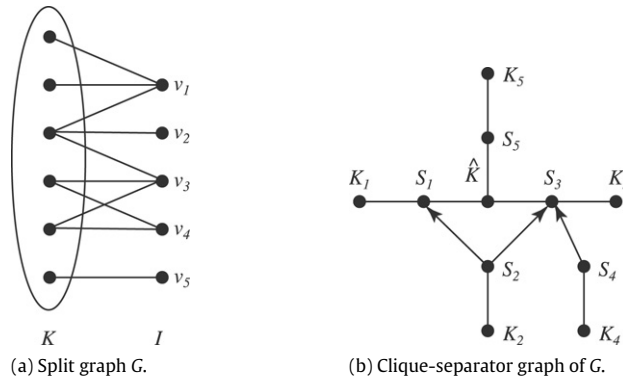


Fig. 11. Example of Theorem 11(1).

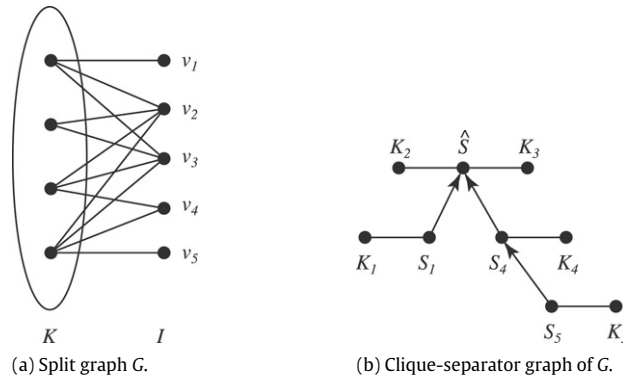


Fig. 12. Example of Theorem 11(2).

v and $\{u, v\} \notin E$.) Split graphs are a proper subclass of chordal graphs and are not comparable to interval graphs or proper interval graphs.

The next theorem presents the properties of \mathcal{G} when G is a split graph. Fig. 11 illustrates Property 1 of the theorem with clique node $\hat{K} = K$ and Fig. 12 illustrates Property 2 of the theorem with separator node $\hat{S} = K$. In both figures, each $K_i = N(v_i) \cup \{v_i\}$ and each $S_i = N(v_i)$, where $N(v)$ is the set of neighbors of vertex v . A vertex is isolated if it has degree zero.

Theorem 11. Let G be a chordal graph with clique-separator graph \mathcal{G} . Then G is a split graph if and only if \mathcal{G} has either of the following properties.

- (1) There is a clique node \hat{K} such that every separator node is a predecessor of \hat{K} and every other clique node K is a leaf that contains exactly one more vertex than its neighbor, unless K consists of one isolated vertex.
- (2) There is a separator node \hat{S} such that every other separator node is a predecessor of \hat{S} and every clique node K is a leaf that contains exactly one more vertex than its neighbor, unless K consists of one isolated vertex.

Proof. (\Leftarrow) Suppose \mathcal{G} has Property 1. If \mathcal{G} has exactly one clique node, then G is a complete graph and we are done. Otherwise, if every clique node of \mathcal{G} contains exactly one vertex, then G is an independent set and we are done. Otherwise, let K_1, \dots, K_l be the clique nodes that contain more than one vertex and let S_1, \dots, S_l be their neighbors, respectively. Each K_i contains exactly one vertex v_i not contained in \hat{K} . Since each K_i is a leaf, S_i divides K_i and K_j for any $i \neq j$. Then by Corollary 4(1), $\{v_i \mid 1 \leq i \leq l\}$ is an independent set. Therefore, the vertex set of G is partitioned into clique \hat{K} and independent set $\{v_i \mid 1 \leq i \leq l\} \cup \{\text{isolated vertices}\}$. Suppose \mathcal{G} has Property 2. Then a similar argument shows that the vertex set of G is partitioned into clique \hat{S} and independent set $\{v_i \mid 1 \leq i \leq l\} \cup \{\text{isolated vertices}\}$.

(\Rightarrow) Suppose the vertex set of G can be partitioned into clique K and independent set I . Observe that for every $v \in I$, $N(v) \cup \{v\}$ is a maximal clique, and that K is not a maximal clique if and only if some vertex in I is adjacent to every vertex in K .

Case 1: K is a maximal clique. Consider any $v \in I$. If $N(v) = \emptyset$, then v is an isolated vertex. Otherwise, $N(v) \subset K$ and $N(v)$ is a minimal uv -separator for every $u \in K - N(v)$, which means $N(v)$ is a minimal vertex separator. Then in \mathcal{G} , separator node $N(v)$ and clique node $N(v) \cup \{v\}$ are adjacent and $N(v)$ is a predecessor of clique node K . It follows that \mathcal{G} has Property 1 with $\hat{K} = K$. Case 2: K is not a maximal clique. Let v_1, \dots, v_j be the vertices of I that are adjacent to every vertex of K , $j \geq 1$, and let v_{j+1}, \dots, v_k be the remaining vertices of I . If $j = 1$, then for each $j + 1 \leq i \leq k$, $N(v_i) \subset K$ and, as before, separator

node $N(v_i)$ and clique node $N(v_i) \cup \{v_i\}$ are adjacent and $N(v_i)$ is a predecessor of clique node $K \cup \{v_1\}$. Thus, \mathcal{G} has Property 1 with $\hat{K} = K \cup \{v_1\}$. If $j > 1$, then K is a separator node adjacent to each of the clique nodes $K \cup \{v_1\}$, $K \cup \{v_2\}$, \dots , $K \cup \{v_j\}$. For each $j + 1 \leq i \leq k$, $N(v_i) \subset K$ and, as before, separator node $N(v_i)$ and clique node $N(v_i) \cup \{v_i\}$ are adjacent and $N(v_i)$ is a predecessor of separator node K . It follows that \mathcal{G} has Property 2 with $\hat{S} = K$. \square

7. Computing the clique-separator graph

There are several algorithms that recognize a chordal graph and, if it is chordal, compute a clique tree of the graph. Tarjan and Yannakakis [25] gave an algorithm that recognizes chordal graphs and implicitly computes a clique tree and the maximal cliques of the graph. Lewis, Peyton, and Pothen [18] gave an algorithm that computes the same explicitly, and this algorithm was simplified by Blair and Peyton [2]. Each of these algorithms runs in $\Theta(m + n)$ time.

Given a chordal graph G , the following algorithm computes the separator nodes and arcs of \mathcal{G} . We will later show how to compute the clique nodes and edges of \mathcal{G} . Recall that several edges of a clique tree of G may correspond to the same minimal vertex separator of G .

- (1) [Preprocess.] Verify that G is chordal and compute a clique tree T of G . Compute the minimal vertex separators S_1, S_2, \dots, S_p of G , where each $S_i = K \cap K'$ and $\{K, K'\}$ is the i th edge of T . (Each S_i can be represented by a linked list or characteristic vector.)
- (2) [Compare S_i s.] Create an $n \times n$ 0–1 matrix M and initialize it to 0. For every $1 \leq i < j \leq p$, compare S_i and S_j . If $S_i = S_j$, then delete S_j . If $S_i \subset S_j$, then set $M[i, j]$ to 1. If $S_j \subset S_i$, then set $M[j, i]$ to 1.
- (3) [Create nodes and arcs.] Sort the S_i s by size. Create a separator node for every S_i . Then do the following.
 - For $l = n$ downto 2.
 - For $k = l - 1$ downto 1.
 - For every S_i of size k and every S_j of size l , if $M[i, j] = 1$ and $M[i, j'] = 0$ for every $S_{j'}$ such that $(S_{j'}, S_j)$ is an arc, then create arc (S_i, S_j) .

Step 1 runs in $O(n^2)$ time since computing a clique tree of G requires $\Theta(m + n)$ time. Step 2 runs in $O(n^3)$ time since there are at most n S_i s, each with size at most n , and so comparing any S_i, S_j pair requires $O(n)$ time. Step 3 runs in $O(n^3)$ time because the if–then statement tests each S_i, S_j pair at most once and a separator node has at most n immediate predecessors. Thus, the algorithm runs in $O(n^3)$ time.

We now show the algorithm correctly computes the arcs of \mathcal{G} by showing that Step 3 creates arc (S_i, S_j) if and only if $S_i \subset S_j$ and there is no $S_{i'}$ such that $S_i \subset S_{i'} \subset S_j$. Consider iteration $l = n$ of the l loop and let S_j be a separator node of size n . Consider the k loop. An induction argument shows that each iteration $k = k'$ creates arc (S_i, S_j) for every S_i with size k' such that $S_i \subset S_j$ and no separator node $S_{i'}$ satisfies $S_i \subset S_{i'} \subset S_j$. (If $S_{i'}$ exists, then a previous iteration of the k loop created either arc $(S_{i'}, S_j)$ or arc $(S_{j'}, S_j)$ for some $S_{j'}$ where $S_{j'} \subset S_j$.) Therefore, the arcs into every separator node of size n are correctly computed. The same argument shows that after the remaining iterations of the l loop, the arcs into every separator node are correctly computed. Thus, the algorithm correctly computes the arcs of \mathcal{G} .

To compute the clique nodes and edges of \mathcal{G} , we extend the algorithm so that the maximal cliques are processed in the same way and at the same time as the minimal vertex separators. There are at most n maximal cliques and so M is now a $2n \times 2n$ matrix. After the algorithm halts, we replace every arc from a separator node to a clique node with an edge. Hence, the given algorithm computes the clique-separator graph of a chordal graph in $O(n^3)$ time.

Next, we show that the running time can be improved when G is an interval graph; the faster algorithm computes the clique-separator graph of G from a clique path of G . There are numerous algorithms that recognize an interval graph and, if it is an interval graph, compute a clique path of the graph. Booth and Lueker [3] gave the first linear time algorithm and Corneil, Olariu, and Stewart [6] gave a much simpler algorithm, which computes a clique path and an interval representation of G . Both algorithms run in $\Theta(m + n)$ time. Given an interval representation of G , let $I(v)$ be the interval assigned to vertex v and let $\text{Intersect}(S)$ be the intersection of the intervals assigned to the vertices in set S .

Theorem 12. *Let G be an interval graph with clique-separator graph \mathcal{G} . For any separator node S and any node N of \mathcal{G} , $S \subset N$ if and only if $\text{Intersect}(N) \subset \text{Intersect}(S)$.*

Proof. (\Rightarrow) Suppose $S \subset N$. Then $\text{Intersect}(N) \subseteq \text{Intersect}(S)$ and we must show that $\text{Intersect}(N) \subset \text{Intersect}(S)$. Since $S \rightsquigarrow N$, it follows that N is in some connected component of $\mathcal{G} - \text{Preds}(S)$. By Corollary 4(4), $S \rightsquigarrow K$ for some clique node K in a different connected component of $\mathcal{G} - \text{Preds}(S)$. Then S divides N and K . By Corollary 4(1), S separates any vertex $u \in N - S$ and any vertex $v \in K - S$. Now if $\text{Intersect}(N) = \text{Intersect}(S)$, then $I(u)$ contains $\text{Intersect}(S)$ (because $\text{Intersect}(N) \subseteq I(u)$) and $I(v)$ intersects $\text{Intersect}(S)$ (because v is adjacent to every vertex in S). But then $I(u)$ intersects $I(v)$ and thus u is adjacent to v , a contradiction. Therefore, $\text{Intersect}(N) \subset \text{Intersect}(S)$.

(\Leftarrow) Since G is an interval graph, it has a clique path (K_1, K_2, \dots, K_l) . The maximal cliques of G define a sequence of intervals $\text{Intersect}(K_1), \text{Intersect}(K_2), \dots, \text{Intersect}(K_l)$ that are pairwise disjoint (because if $\text{Intersect}(K_i)$ intersects $\text{Intersect}(K_j)$, then every vertex in $K_i - K_j$ is adjacent to every vertex in K_j and thus K_j is not maximal, a contradiction). By Theorem 1(1), each $S_i = K_i \cap K_{i+1}$, $1 \leq i < l$, is a minimal vertex separator of G and each minimal vertex separator of G is some S_i .

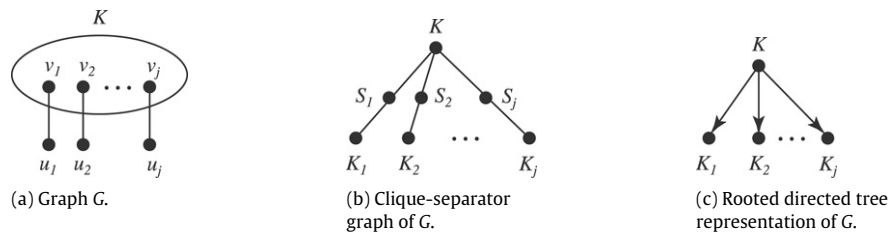


Fig. 13. Example.

Suppose $\text{Intersect}(N) \subset \text{Intersect}(S)$. Case 1: N is a clique node. If $S \not\subseteq N$, then every vertex in $S - N$ is adjacent to every vertex in N , a contradiction. Since a minimal vertex separator cannot be a maximal clique, $S \subset N$. Case 2: N is a separator node. Let $N = S_i$ and $S = S_j$. Since every vertex in S_i is adjacent to every vertex in K_i and every vertex in K_{i+1} , $\text{Intersect}(S_i)$ intersects $\text{Intersect}(K_i)$ and $\text{Intersect}(K_{i+1})$. Since $\text{Intersect}(S_i) \subset \text{Intersect}(S_j)$, it follows that $\text{Intersect}(S_j)$ intersects $\text{Intersect}(K_i)$ and $\text{Intersect}(K_{i+1})$. Then every vertex of S_j is adjacent to every vertex in K_i and every vertex in K_{i+1} . Since K_i and K_{i+1} are maximal cliques, $S_j \subseteq K_i$ and $S_j \subseteq K_{i+1}$, which means $S_j \subseteq K_i \cap K_{i+1} = S_i$. Since $\text{Intersect}(S_j)$ is a smaller interval than $\text{Intersect}(S_j)$, we must have $S_j \subset S_i$, i.e., $S \subset N$. \square

Theorem 12 allows us to improve the running time of the algorithm to compute \mathcal{G} when G is an interval graph, as follows. In Step 1, we compute a clique path and an interval representation of G in $\Theta(m + n)$ time. Since comparing any S_i, S_j pair requires constant time, matrix M is unnecessary and Step 2 is omitted. Step 3 runs in $O(n^2)$ time because by [Lemma 5\(2\)](#), a separator node has at most two immediate predecessors. Hence, the given algorithm computes the clique-separator graph of an interval graph in $O(n^2)$ time.

Lastly, we show that the running time can be improved further when G is a proper interval graph. By [Corollary 10](#), G has a unique clique tree, which is a path P . By [Lemma 9](#), the family of minimal vertex separators corresponding to the edges of P is an antichain. Therefore, inserting the minimal vertex separator $K \cap K'$ between every consecutive pair K and K' of maximal cliques in P yields the only box of \mathcal{G} , as shown in the proof of [Corollary 10](#). Hence, we can compute the clique-separator graph of a proper interval graph in $\Theta(m + n)$ time.

8. Other subclasses of chordal graphs

In the *intersection graph* of a family of sets, each vertex corresponds to a set in the family and two vertices are adjacent exactly when their corresponding sets have a nonempty intersection. Thus, an interval graph is the intersection graph of a family of intervals and a proper interval graph is the intersection graph of a family of inclusion-free intervals. A chordal graph is the intersection graph of a family of subtrees of a tree [[5,11,27](#)] and a split graph is the intersection graph of a family of distinct subtrees of a star [[21](#)], where each subtree is considered to be a set of vertices.

In [Section 4](#), we showed that the clique-separator graphs of interval graphs have structural properties not shared by the clique-separator graphs of chordal graphs. In this section, we examine whether interval graphs are contained in a larger class that does share these structural properties. In particular, we want to know whether interval graphs are properly contained in a well-known subclass \mathcal{C} of chordal graphs such that the clique-separator graphs of graphs in \mathcal{C} have the structural properties in [Lemma 5\(2\)](#) and [Lemma 6\(2\)](#).

One candidate for \mathcal{C} is the RDV graphs. A *rooted directed vertex path* (RDV) graph is the intersection graph of a family of directed paths in an rooted directed tree (the root is the only vertex with in-degree zero, or equivalently, all edges are directed away from the root), where each path is considered to be a set of vertices. We have $\{\text{interval graphs}\} \subseteq \{\text{RDV graphs}\}$. Johnson [[17](#)] refers to RDV graphs as *directed path graphs* in his discussion of “algorithmically significant classes of intersection graphs”. Monma and Wei [[22](#)] present a unified framework for studying intersection graphs of families of paths in a tree and they characterize RDV graphs as well as larger subclasses of chordal graphs.

[Fig. 13](#) shows (a) a chordal graph G with maximal cliques $K = \{v_1, v_2, \dots, v_j\}$ and K_1, K_2, \dots, K_j , where each $K_i = \{u_i, v_i\}$, (b) the clique-separator graph of G , (c) a rooted directed tree representation of G where each v_i corresponds to path (K, K_i) and each u_i corresponds to path (K_i) . Thus, G is an RDV graph whose clique-separator graph contains a clique node with j neighbors. Now suppose we modify G by adding vertices x and y that are adjacent to every vertex in K and nonadjacent to each other. Then $S = \{v_1, v_2, \dots, v_j\}$ is a minimal vertex separator and $K' = S \cup \{x\}$ and $K'' = S \cup \{y\}$ are maximal cliques. We can readily show that the modified G is an RDV graph whose clique-separator graph contains a separator node with j immediate predecessors. Now the clique-separator graph in [Fig. 13\(b\)](#) is a chordal graph. We can readily show that this graph is an RDV graph whose clique-separator graph contains a separator node with j neighbors that are internal nodes of the box containing them. Therefore, the clique-separator graphs of RDV graphs do not have any of the structural properties in [Lemma 5\(2\)](#) and [Lemma 6\(2\)](#). (These examples also show that $\{\text{interval graphs}\} \subset \{\text{RDV graphs}\}$.)

Johnson [[17](#)] discusses two more well-known subclasses of chordal graphs: *strongly chordal graphs* and *undirected path graphs*. Since the RDV graphs are contained in each of these subclasses [[17](#)], their clique-separator graphs do not have the structural properties in [Lemma 5\(2\)](#) and [Lemma 6\(2\)](#). We remark that the chordal graphs discussed in the previous paragraph

are ptolemaic graphs [26], so their clique-separator graphs also do not have the structural properties in Lemma 5(2) and Lemma 6(2).

It would be interesting to know whether the clique-separator graphs of one of these subclasses have other properties that the clique-separator graphs of chordal graphs do not have, or to characterize them in terms of the clique-separator graph. Similarly, it would be interesting to study the clique-separator graphs of other subclasses of chordal graphs, such as chordal graphs that are comparability graphs.

References

- [1] P.A. Bernstein, N. Goodman, Power of natural semijoins, *SIAM J. Computing* 10 (1981) 751–771.
- [2] J.R.S. Blair, B. Peyton, An introduction to chordal graphs and clique trees, in: *Graph Theory and Sparse Matrix Computation*, in: IMA, vol. 56, Springer-Verlag, New York, 1993, pp. 1–29.
- [3] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *J. Comput. System Sci.* 13 (1976) 335–379.
- [4] A. Brandstädt, V.B. Le, J.P. Spinrad, *Graph classes: A survey*, Society for Industrial and Applied Mathematics, Philadelphia, 1999, Monograph.
- [5] P. Buneman, A characterization of rigid circuit graphs, *Discrete Math.* 9 (1974) 205–212.
- [6] D.G. Corneil, S. Olariu, L. Stewart, The ultimate interval graph recognition algorithm? in: *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 175–180.
- [7] G.A. Dirac, On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg* 25 (1961) 71–76.
- [8] S. Földes, P.L. Hammer, Split graphs, *Congr. Numer.* 19 (1977) 311–315.
- [9] D. Fulkerson, O. Gross, Incidence matrices and interval graphs, *Pacific J. Math.* 15 (3) (1965) 835–855.
- [10] P. Galinier, M. Habib, C. Paul, Chordal graphs and their clique graphs, in: *Proceedings of the 21th International Workshop on Graph Theoretic Concepts in Computer Science*, in: LNCS, vol. 1017, 1995, pp. 358–371.
- [11] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *J. Combin. Theory Ser. B* 16 (1974) 47–56.
- [12] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, in: *Annals of discrete mathematics*, vol. 57, Elsevier B.V., Amsterdam, 2004.
- [13] P. Hell, R. Shamir, R. Sharan, A fully dynamic algorithm for recognizing and representing proper interval graphs, *SIAM J. Computing* 31 (2001) 289–305.
- [14] C. Ho, R.C.T. Lee, Counting clique trees and computing perfect elimination schemes in parallel, *Inform. Process. Lett.* 31 (1989) 61–68.
- [15] L. Ibarra, A fully dynamic graph algorithm for recognizing interval graphs, *Algorithmica* (2009) (in press).
- [16] L. Ibarra, A fully dynamic graph algorithm for recognizing proper interval graphs, 2008 (submitted for publication).
- [17] D.S. Johnson, The NP-completeness column: an ongoing guide, *J. Algorithms* 6 (1985) 434–451.
- [18] J.G. Lewis, B.W. Peyton, A. Pothén, A fast algorithm for reordering sparse matrices for parallel factorization, *SIAM J. Computing* 10 (6) (1989) 1146–1173.
- [19] M. Lundquist, Zero patterns, chordal graphs, and matrix completions, Dept. of Mathematical Sciences, Clemson University, 1990.
- [20] T.A. McKee, F.R. McMorris, *Topics in intersection graph theory*, Society for Industrial and Applied Mathematics, Philadelphia, 1999, Monograph.
- [21] F.R. McMorris, D.R. Shier, Representing chordal graphs on $K_{1,n}$, *Comment. Math. Univ. Carolin.* 24 (1983) 489–494.
- [22] C.L. Monma, V.K. Wei, Intersection graphs of paths in a tree, *J. Combin. Theory Ser. B* 41 (1986) 141–181.
- [23] F.S. Roberts, Indifference graphs, in: F. Harary (Ed.), *Proof Techniques in Graph Theory*, Academic Press, New York, 1969, pp. 139–146.
- [24] Y. Shibata, On the tree representation of chordal graphs, *J. Graph Theory* 12 (3) (1988) 421–428.
- [25] R.E. Tarjan, M. Yannakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, *SIAM J. Computing* 13 (3) (1984) 566–579.
- [26] R. Uehara, Y. Uno, Laminar structure of ptolemaic graphs and its applications, in: *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, in: *Lecture Notes in Computer Science*, vol. 3827, 2005, pp. 186–195.
- [27] J.R. Walter, Representations of chordal graphs as subtrees of a tree, *J. Graph Theory* 2 (1978) 265–267.